SYSTEMS AND METHODS FOR PREVENTING UNAUTHORIZED COPYING OF SOFTWARE DOWNLOADED FROM A REMOTE SERVER

TECHNICAL FIELD OF THE INVENTION

This invention relates generally to systems and methods for preventing the piracy of computer software, and more particularly to systems and methods for preventing the unauthorized distribution of computer software downloaded over a distributed network.

BACKGROUND

Downloading software application program over the Internet provides software developers with a cost-effective alternative for distributing their products. Instead of burning their software product onto a magnetic media, such as a diskette, or onto an optical media, such as a compact disk ("CD"), software developers can simply place their software on a web site and allow users to download the software program directly onto their computer.

After the user has selected and paid for the software application program, the user will typically accept a "click wrap" license agreement by selecting an "I Agree" button, or some similar button, at the end of the license agreement. Typically, the license agreement limits the user to install the software on a single computer and prohibits the user from distributing the software application program to anyone else. These licensing agreements are known as single-user licensing agreements and are generally meant for the typical consumer.

Fortunately, the majority of people who purchase software application programs and other software products over the Internet, respect the license agreement and use the software in accordance with the license agreement. Unfortunately, for one reason or another, many people do not respect the terms of the licensing agreement. For example, "computer hackers" disregard the license agreement and freely distribute unauthorized copies of the software product once they have downloaded it from the Internet. The hacker may be financially motivated to sell the unauthorized pirated copies of the product to others. The hacker may not receive any financial gain, but nonetheless may be motivated to freely distribute the software product, such as by uploading the product onto a news server in which any computer user can freely access and download the program without paying for it. The distribution of the software may be less malevolent, such as a child computer user who may not fully comprehend the terms of the agreement and who distributes several unauthorized copies to his friends. Regardless of the reason, software piracy poses a substantial cost to the software developer.

However, there is no easy way for the software developer to monitor whether users are complying with the license agreement. Several methods have been developed to curtail the unauthorized distribution of computer software programs. One method involves a user of a software application program to register with a central server. Once the user is registered a critical portion of the application is downloaded from the central server to the user's computer. The critical code is encrypted with some of the user's personal information, such as name, address, birthday, etc. as well as the computer serial number. Each time the software program

is executed, the code is checked against a user identification code to ensure that the code has not been tampered with. If the codes do not match, the program will not execute. Unfortunately, this method will only work on a single machine. The method can be defeated by copying the entire contents of the computer's hard drive to another machine. Furthermore, this method does not allow for tracking the distribution of unauthorized copies of the software application program.

Several anti-piracy software methods involve checking the validation of the software application program by accessing a database on a central server over a distributed network, such as the Internet. In one method, if the software application program does not have a valid license, the software application program is disabled. In another method the number of registered copies of a software application program running on a network is tracked. Each time a copy of the software application program is executed a counter is incremented and checked against a central database that contains the number of licenses for that software application program. If the counter does not exceed the number of number of allowed licenses, the software application program is executed over the network. If, however, the counter exceeds the number of allowed licenses stored in the central server, then the execution of the software application program is terminated. Each of these methods has several limitations. First, each method is limited to software applications running on a distributed network. The method only checks to insure that the software program running on the network has a valid license. There is no way to check for software application programs that run independently on stand-alone computer for valid licensees. Furthermore, there is no way for these methods to track any unauthorized

copies of the software program back to the user who distributed the unauthorized software copies.

Thus, there is a need in the art for methods and systems that prevent the unauthorized distribution of software application program. There is a further need in the art for methods and systems for tracking the distribution of unauthorized copies of a software application program in the event that a hacker is able to distribute the unauthorized copies of the software application program. Being able to trace each unauthorized copy back to the individual who distributed the unauthorized copies allows the individual to be charged and/or fined in accordance with the law.

SUMMARY OF THE INVENTION

The invention addresses the problems above by providing methods and systems for preventing unauthorized distribution of software application programs that are downloaded over a distributed network. Generally described, the invention is a method that prohibits the distribution of unauthorized copies of the software application program in violation of the license agreement. The method begins when a user first purchases the software application program for downloading to their computer from an application server platform. Once the user purchases the software and agrees to the license agreement, an installation program, also known, as a "Client Program" is first downloaded from the application server platform to the user's computer over a non-secured communications channel. Next, the user executes the Client Program, which in turn contacts the server platform and establishes a second communications channel with the server platform. The Client Program also scans the

user's entire computer system to determine the identification numbers of the various components installed on the user's computer. The identification numbers are typically but not limited to the unique class identifiers associated with each hardware driver for each component installed on the user's machine. The number of class identifiers varies from computer to computer. The Client Program uses these class identifiers to generate a unique user identifier ("UUI") code that is exclusive to the user's machine. The Client Program then transmits the UUI over the encrypted communications channel to the server platform, where it is broken into several smaller segments and embedded in various locations in several files associated with the software application program. The files are downloaded to the user's computer and installed at various locations throughout the hard drive of the user's computer.

Each time the software application is executed, the client program, runs an anti-piracy check, which determines whether the UUI of the user's machine matches the UUI embedded in the files of the software application program. If the UUI of the user's machine does not match the UUI embedded within the software's file, the execution of the software program terminates. If, however, the UUI of the machine matches the UUI embedded within the files, the software application program will execute normally and the user will never know that the anti-piracy check was performed.

Other advantages and features of the invention will be apparent from the description below, and from the accompanying papers forming this application.

BRIEF DESCRIPTION OF DRAWINGS

The accompanying drawings, which are incorporated in and form a part of the specification, illustrate preferred embodiments of the present invention and, together with the description, disclose the principles of the invention.

FIG. 1 is a block diagram of a prior art method for downloading computer software over a distributed network.

FIG. 2 is a block diagram of a method of downloading computer software over the Internet to prevent unauthorized copying of the computer software.

FIG. 3 is a block diagram of a method of tracking the unauthorized distribution of computer software using the present invention.

FIGs. 4A and 4B, hereinafter collectively referred to as FIG. 4, are logic flow diagrams illustrating a method for downloading computer software over the Internet using the present invention.

DETAILED DESCRIPTION

FIG. 1 is a block diagram illustrating a prior art method of downloading a software application program from the Internet and allowing for the unauthorized distribution of the downloaded software. To download a software application program over the Internet, an authorized user, through his or her computer 105, accesses a remote server 110 over a first communications channel 115 of a distributed network, such as the Internet, where the software application program is located. Typically, the first communications channel 115 will be a non-secured communications channel because no sensitive information is being transmitted

between the server platform and the client platform. After the authorized user computer 105 connects with the remote server 110, the user is presented with a software licensing agreement for the software application program that he or she wishes to download.

After the user signifies his or her agreement to the terms of the license agreement, usually by selecting and "I Agree" button at the bottom of the license agreement, the software application program is downloaded over a second communications channel 120 to the authorized user's client platform 105. Typically, the second communications channel is a secured channel to minimize the possibility of "hackers" tapping the communications link and illegally obtaining the application program. Typically, the files associated with the software application program are compressed into a single file, known as "zip" file, to reduce space and downloading time. Once the computer program is successfully downloaded to the user's client platform 105, the application program is installed. At this point most users will use the software application program in accordance with the license agreement, that is, they will only use the software application program on one machine. However, there are some authorized users that willingly violate the software license agreement by distributing unauthorized copies of the software application program to an unauthorized user who installs it on an unauthorized platform 125. As an example, a teenage user may legally download a computer game from a remote server after agreeing to the licensing agreement, which prohibit distributing the software to third parties. However, after playing the computer game, the teenage user makes a copy of the downloaded computer game and distributes it to his friends. Unfortunately, by

making a copy of the downloaded computer game and distributing it to his friend, the teenage user has violated the software license agreement and more importantly, the manufacture of the computer game has lost a potential sale. Furthermore, the manufacture of the software game has no way to track to whom the teenage user has distributed unauthorized copies.

A more serious occurrence of distributing unauthorized copies of software application program occurs when the authorized user uploads the software application program to a remote server, such as a news server 140, which is accessible to everyone on the Internet. Once the software application program is uploaded to news server 140, it can be freely accessed as an Internet software download 145 by any unauthorized users (155A, 155B, ... 155N) attached to the Internet. In this manner, the software application program can be distributed to an unlimited number of unauthorized users, which can result in untold losses to the software manufacture.

Reference will now be made in detail to preferred embodiments of the invention, non-limiting examples of which are illustrated in the accompanying drawings.

FIG. 2 is a block diagram illustrating a distributed network for downloading a software application program over a distributed network in accordance with the present invention. The software application program is stored on an application server 110. The process begins when a user purchases the software application program by accessing the application server 110 from a client platform 105, over the distributed network. Typically, the client platform 105 will be a personal computer, however, those skilled in the art will appreciate that other devices,

such as personal digital assistants (PDAs) cellular telephone devices, pagers, MP3 players, and the like may be used as the personal platform 105. In purchasing the software application program the authorized user must agree to a "click-wrap" software license agreement, which typically limits the user to load and execute the software application program on a single client platform. Once the user has purchased the software application program, a software application program, called a Client Program, is downloaded from a server onto the client platform, hard drive. Also, purchase of the software application program, along with an identifying indicia associated with the Client Program is recorded on a remote server 205 for registration purposes.

The user then executes the Client Program, which automatically installs itself on the client platform 105. Alternatively, the Client Program may be a self-executing program, which automatically executes once it is downloaded onto the client platform 150. Once the Client Program has been executed, the Client Program automatically generates a unique user identifier (UUI) 217 that is unique to the client platform 105. To create the UUI 217, the Client Program first retrieves identifying indicia from the client platform. Normally, the identifying may be the global unique identifications (GUIDs) of each device installed on the client platform 105, the network interface card media access control (NIC MAC), the ID of the hard drive, or the LBA of every immoveable file installed on the client platform 105, or any other identifiers attached to the client platform 105. Typically, the Client Program collects the identifying indicial in a predetermined order. First, the Client Program retrieves the GUID for each device attached to the client platform then the NIC MAC, then the hard drive ID

and then the immoveable files LBA. However, then number of unique class identifiers will vary from client platform to client platform. This provides an advantage in that each UUI 217 will be distinctive of the particular client platform 105. However, the Client Program may be customized to include a single identifying indicia or as many identifying indicia as requested by the software manufacturer. Additional security may be added by included a unique customer identification number and/or unique BIOS information form the client platform 105. The UUI 217 is typically formatted as a delimited hexadecimal string.

The Client Program then establishes an encrypted communications channel 120 with the application server 110. The Client Program then uploads the UUI 217 to the application server 110 over the encrypted communications channel 120. An encrypted channel 120 is used to upload the UUI 217 to reduce the risk that "hackers" will steal the UUI code. In addition to the UUI, the identifying indicia associated with the Client Program are uploaded to the application server 110, which in turn transmits the identifying indicia to the remote server 205. The identifying indicia are compared to a database to determine whether the Client Program has been activated. If the determination is made that the Client Program has not been previously executed, a first command is transmitted back to the application server indicating that the Client Program has not been previously executed and the downloading of the software application program is allowed to proceed. A flag is then set in the database indicating that that particular Client Program has been executed.

If, however, the determination is made that the Client Program has been previously executed, a second command is transmitted to the application server 205,

which "locks out" the software application program and prohibits the software application program from being downloaded from the application server 110 to the client platform 105.

Once the determination is made that the Client Program has not been previously executed, the UUI is then passed to a control program that separates the UUI into several segments. Typically, the control program breaks the UUI into segments by MD5 hashing the UUI and dividing it into a series of 16 bit words.

After the UUI is hashed into segments, each segment is replicated multiple times. The number of times each segment is replicated is variable and is determined by the software manufacturer. For instance, one software manufacturer may distribute a fairly small and inexpensive program that they feel will not be widely pirated and require that each segment of the hashed UUI is replicated twice. Contrarily, another software manufacturer may distribute a costly and complex program that they fear may be widely copied. In this instance the manufacturer may request that each segment of the hashed UUI be replicated 10 times to assure themselves that a "hacker" will not be able to eliminate each instance of the UUI and freely distribute unauthorized copies of the software application program. By having the flexibility to allow each manufacture to set the number of times the hashed UUI is replicated insures that each manufacturer is comfortable with the individual level of security they believe is necessary to stop the unauthorized distribution of their software application programs.

Each instance of the hashed segments are then embedded into various locations throughout the files of the software application program. Each file is then

downloaded over the encrypted channel 120 and installed in different locations on the client platform's memory storage device. For example, if the client platform is a personal computer, the memory storage device will typically be a hard drive. If, however, the client platform 105 is a PDA or a web enable cellular telephone, the memory storage device may be a memory card. Downloading the software application program files over the encrypted channel 120 ensures that not only a "hacker," but also the authorized user cannot see where the files are being placed on the memory storage device. Preventing the authorized user from knowing the location of the software application program files on the memory storage device not only makes it impossible for the typical user, but also makes it difficult for the experienced hacker to distribute unauthorized copies.

Once all the files of the software application program have been placed in the memory storage device of the client platform 105, the software application program is ready to be executed by the authorized user. Each time the authorized user selects the software application program, the software application program launches the Client Program. The Client Program determines the UUI 217 of the client platform 105 and checks it against the UUI 217 embedded within the software application program files. If, the UUI 217 of the client platform 105 matches the UUI 217 embedded within the software application program, the software application program executes in the usual manner without the user ever having known that the UUI 217 was checked. If, however, the UUI 217 of the client platform 105 does not match the UUI 217 embedded in the software application program, the Client Program prevents the software application program from executing.

As an example, suppose the authorized user attempts to give an unauthorized copy of the software application program to a friend who is an unauthorized user, by transmitting the Client Program from the client platform 105 to the unauthorized user's platform 125 over a communications link 130 and the unauthorized user attempts to run the Client Program. The Client Program will contact the application server 110, and transmit the machine's UUI 217 and the identifying indicia associated with the Client Program. The application server 110, in turn transmits the identifying indicia associated with the Client Program to the remote server, where it is checked against the database. Since the Client Program has been previously executed by the authorized user, the remote sever 205 will transmit the second command back to the application server 110, which prevents the software application from being downloaded to the unauthorized user's platform 125.

If, however, the authorized user is an experienced hacker and has experience with pirating software, the authorized user may be able to locate all of the files installed and used by the software application program. If the authorized user is able to locate each file, the authorized user may then be able to bundle the files into a single file that he may distribute to any number of unauthorized users. Even after an unauthorized user downloads this single file and manages to correctly install it on their platform 125, the software would still not work. This is because the unauthorized user's platform 125 would have a different UUI 217. Thus, when the Client Program calculates the UUI of the unauthorized user's platform 125 and compares it to the UUI 217 embedded throughout the files of the software application

program, the two UUI's 217 would not match and the Client Program would terminate the execution of the software application program.

FIG. 3 is a block diagram illustrating the method of tracking unauthorized copies of the software application program. An authorized user downloads an authorized copy of the software application program from the application server 110 to a client platform 105 as described above. The software application program has multiple instances of the UUI 217 embedded within its files and is lawfully installed on the client platform 105 in accordance with the license agreement. Suppose however, that the authorized user is an exceptionally talented hacker and is able to decompile the software application program executable and examine the encoded data to determine where within the files the UUI 217 is checked against the UUI 217 of the host platform. The hacker then deletes the UUI checks so that when the software application program is executed, the UUI checks are no longer performed. The hacker then believes that he or she can distribute unauthorized copies of the software application program to unauthorized user platforms 125 through the communications channel 130. The easiest method of distributing pirated software is for the hacker to make CD copies of the software application program and give the CDs to his or her friends. Alternatively, on a broader and more egregious scale, the authorized user could post the software application program to a news server 140 so that any user operating a platform (155A, 155B, ... 155N) with a connection to the Internet 145 could have access to the software application program. Any user (155A, 155B, ... 155N) could download the pirated version of the software application program over the Internet 145 and run it uninhibited on their computer. However, unknown to the

authorized user is that the multiple instances of the UUI 217, typically the authorized users client platform 105, is embedded within the files of the software application program. Because the UUI 217 is embedded in locations within the files that are not used to check to determine whether the software application program is pirated, the hacker is unaware that the UUI 217 is transmitted with each copy of the pirated software. Therefore, by embedding multiple instances of the UUI 217 within the files of the software application program, unauthorized copies of the software application program can be tracked so that the unauthorized users can be arrested and fined under the appropriate laws.

FIGs. 4A and 4B are logic flow diagrams illustrating a routine 400 for downloading a software application program over the Internet using the present invention. The routine 400 begins at step 405, in which the user purchases the software application program from a vendor over a distributed network, such as the Internet. The user must establish a connection with an appropriate application server, which stores the desired software application program. In purchasing the software application program, the user typically agrees to a "click wrap" license agreement, which normally states that the user will only install and run the software application program on a single platform.

Once the user aggress to the license agreement, step 405 is followed by step 410, in which a Client Program is downloaded from the application server to the user's client platform. Normally the client platform will be a personal computer. However, the client platform may be a PDA, a web-enabled cellular telephone, or any other device capable of storing and executing a software application program.

Next, step 410 is followed by step 415, in which the user executes the Client Program, which automatically installs itself on the client platform. Alternatively, the Client Program may be a self-executing program, which automatically executes once it is downloaded onto the client platform.

Step 415 is then followed by step 420, in which the Client Program automatically establishes a secure communications link with the application server. The secure communications link is typically an encrypted channel, which ensures that hackers cannot pilfer the UUI as it is uploaded to the application server.

Next, step 420 is followed by step 425, in which the Client Program automatically generates a unique user identifier (UUI) that is unique to the client platform by retrieving identifying indicia from the client platform. Normally, the identifying indicia may be the global unique identifications (GUIDs) of each device installed on the client platform, the (NIC MAC), the ID of the hard drive, or the LBA of every immoveable file installed on the client platform, or any other identifiers attached to the client platform. The identifying indicia are then combined as a delimited hexadecimal string to form the UUI.

Step 425 is followed by step 430, in which the application server connects with a remote server. The remote server contains a database that stores the client platform's UUI along with a flag that identifies whether the Client Program has been previously executed.

Step 430 is followed by step 435, in which the determination is made whether the Client Program is being executed for the first time. A unique identifier associated with the Client Program is stored in the database on the remote server. Typically, the

unique identifier is a flag that is set when the Client Program is executed. If the flag has not been set, then the Client Program has not been previously executed and the "YES" branch is followed to step 440, in which the Client Program automatically uploads the UUI to the application server over the encrypted channel.

Next, routine 400 proceeds to step 445, in which the UUI information is embedded within multiple files associated with the application program. The UUI is first broken into several 16-bit words by MD5 hashing the delimited hexadecimal string. Each 16-bit word is replicated multiple times and then embedded into the files associated with the software application program. The type of file determines how the 16-bit word is embedded into the file. For executable files, the 16-bit word is embedded within a string placeholder. Image files, on the other hand, are subtly altered to include the UUI into the pixel data. Alternatively, the 16-bit words can be embedded into the image header.

Step 445 is followed by step 450, in which the UUI information is recorded and stored in the database at the remote server. In addition to the UUI information, the location of each instance of each 16-bit word is also stored in the database. This ensures that the UUI can be tracked in the case that a hacker successfully decompile the software application program and delete the UUI checks so that they are no longer performed when the software application program is executed. Knowing the location of each instance of each 16-bit word of the UUI allows the software application program manufacture to compare the files of the suspected unauthorized copy of the software application program with the known locations of the 16-bit words to determine whether the suspected unauthorized copy of the software application

program has been pirated. Additionally, checking the files of the suspected unauthorized copy of the software application program against the know locations of the 16-bit words of the registered UUI allows the software manufactures to trace each pirated copy back to the original authorized user so that the appropriate charges and fines can be levied against the authorized distributing pirated software and also act as a deterrent to other potential hackers from attempting to distribute pirated copies of other software application programs protected in this manner.

Step 450 is then followed by step 455, in which each file of the software application program is downloaded to the client platform over the encrypted channel. The files are then installed in different locations within the memory storage device associated with the client platform so that the software application program will operate correctly. The encrypted channel ensures that the authorized user cannot see where the files are placed with the memory storage device associated with the client platform. Each time the software application program is executed, the UUI of the client platform is checked with the UUI embedded within the files associated with the software application program. If the two UUIs do not match, the software application program terminates. If, however, the two UUIs match, the software application program will execute normally and the UUI check will be transparent to the authorized user. Step 455 is then followed by the "END" step.

Returning the step 435, if the determination is made that the Client Program has previously been executed by checking the flag associated with Client Program stored in the database on the remote server, the "NO" branch is followed to step 460

in which the software application program is terminated. Step 460 is then followed by the "END" step.

It should be understood that the foregoing pertains only to the preferred embodiments of the present invention, and that numerous changes may be made to the embodiments described herein without departing from the spirit and scope of the invention.